

---

---

# Introducción a Ruby on Rails

— Hashes —

---

---

# Overview

- Hashes
  - Cómo son utilizados
  - Por qué son utilizados
  - Similitud con bloques

# Hashes

- Colecciones indexadas de referencias a objetos
  - Creados indistintamente con {} o Hash.new
  - También conocidos como arrays asociativos.
  - El índice (key) puede ser de cualquier tipo.
    - No solamente integer como en los arrays
  - Se accede a un elemento utilizando el operador []
  - Para setear valores:
    - => en la creación
    - [] después de la creación

# Hashes

```
props_editor = { "font" => "Arial", "size" => 12, "color" => "red"}  
# THE ABOVE IS NOT A BLOCK - IT'S A HASH  
puts props_editor.length # => 3  
puts props_editor["font"] # => Arial  
  
props_editor["background"] = "Blue"  
props_editor.each_pair do |key, value|  
  puts "Key: #{key} value: #{value}"  
end  
# => Key: font value: Arial  
# => Key: size value: 12  
# => Key: color value: red  
# => Key: background value: Blue
```

# Hashes

- Que pasa cuando no existe el key?
  - Se retorna **nil**
- Pero si creamos el hash con **Hash.new(0)**
  - Se retorna **0**
- <http://ruby-doc.org/core-2.2.0/Hash.html>

## Hash

A Hash is a dictionary-like collection of unique keys and their values. Also called associative arrays, they are similar to Arrays, but where an Array uses integers as its index, a Hash allows you to use any object type.

Hashes enumerate their values in the order that the corresponding keys were inserted.

A Hash can be easily created by using its implicit form:

```
grades = { "Jane Doe" => 10, "Jim Doe" => 6 }
```

# Ejercicio

Dado el texto: "Oga ari Oga ari oi ka\`i oi ka\`i"

Obtener la frecuencia de aparición de cada palabra.

Por ejemplo: Oga se repite dos veces

# Ejercicio resuelto

```
frecuencia = Hash.new(0)
```

```
sentence = "Oga ari Oga ari oi ka\'i oi ka\'i"
```

```
sentence.split.each do |word|  
  frecuencia[word.downcase] += 1
```

```
end
```

```
p frecuencia # => {"oga"=>2, "ari"=>2, "oi"=>2, "ka'i"=>2}
```

# Más hashes

- Desde **Ruby 1.9**
  - El **orden** de ingreso de los elementos es mantenido.
  - Si se utilizan símbolos como claves se puede utilizar la sintaxis de **simbolos:**
    - simbolo: valor

```
familia_simbolos = {paz: "Blanco", tristeza: "Negro", alegria: "Verde"}  
familia_simbolos[:felicidad] = "Azul"  
p familia_simbolos  
# => {:paz=>"Blanco", :tristeza=>"Negro", :alegria=>"Verde", :felicidad=>"Azul"}
```



# Mas Hashes

- Si un hash es el último argumento de un método se puede obviar el `{}`

```
def ajustar_colores (props = {foreground: "red", background: "white"})
  puts "Foreground: #{props[:foreground]}" if props[:foreground]
  puts "Background: #{props[:background]}" if props[:background]
end

ajustar_colores # => foreground: red
               # => background: white

ajustar_colores ({ :foreground => "green" }) # => foreground: green
ajustar_colores background: "yella" # => background: yella
ajustar_colores :background => "magenta" # => background: magenta
```

# Entonces

Los hashes son colecciones indexadas.

Se usan de forma similar a los arrays.

Pueden ser confundidos con bloques por el uso de {}